N-Queen (8) Problem Analysis Report

Introduction

The n-Queens problem is a classic example in the fields of computer science and artificial intelligence. It involves placing n queens on an n×n chessboard so that no two queens threaten each other. This project aims to solve the n-Queens problem using two distinct local search algorithms: Steepest-Ascent Hill Climbing and a Genetic Algorithm.

Approach

Steepest-Ascent Hill Climbing: This algorithm iteratively moves towards the direction of increasing elevation (or decreasing heuristic cost in our case) to find the optimal solution. It evaluates all neighbors in the current state and moves to the neighbor with the lowest heuristic cost, repeating this process until no better neighbors are found.

Genetic Algorithm: This evolutionary approach starts with a randomly generated population of candidate solutions. It then iteratively applies genetic operators such as selection, crossover, and mutation to evolve the population toward better solutions.

Analysis and Findings

- Implementation: Both algorithms were implemented in Python. The Steepest-Ascent Hill Climbing algorithm evaluated neighbors based on the heuristic of the least number of pairs of queens attacking each other. The Genetic Algorithm used a fitness function, selection, crossover, and mutation processes to evolve the solutions.
- Experimentation:
    - The algorithms were tested over 100 instances of the n-Queens problem.
    - Key metrics like success rate, search cost, and average time taken were recorded.
    - For Steepest-Ascent Hill Climbing, the search cost was measured as the number of neighbor states evaluated, and for the Genetic Algorithm, it was the total number of individuals processed.
- Results:
    - The Steepest-Ascent Hill Climbing algorithm had a low success rate, often failing to find a solution. This was attributed to the algorithm's tendency to get stuck in local optima.
    - The Genetic Algorithm showed a higher success rate, indicating its better exploration capabilities. However, its performance was influenced by randomness and parameters like population size and mutation rate.
    - The Genetic Algorithm tended to have a higher search cost and took longer than Steepest-Ascent Hill Climbing.
    - With 200 test cases:
        - Steepest-Ascent Hill Climbing Success Rate': 14.499999999999998
        - Genetic Algorithm success rate: 22.0
        - Steepest-Ascent Hill Climbing Average Search Cost': 239.68
        - Genetic Algorithm Average Search Cost': 9203.5
        - Steepest-Ascent Hill Climbing Average Time: 0.001863102912902832
        - Genetic Algorithm Average Time: 0.8266999626159668

```
Number of instances: 200
Experiment Results: {'HC_Success_Rate': 14.499999999999998, 'GA_Success_Rate': 22.0, 'HC_Avg_Search_Cost': 239.68, 'GA_Avg_Sear
ch_Cost': 9203.5, 'HC_Avg_Time': 0.00186310291290283 2, 'GA_Avg_Time': 0.8266999626159668}
```

Sample Outputs

Welcome to N-Queen Problem
[1] To run one instance
[2] to run multiple instances
1
Initial Board:

```
. . . . . .
. . Q . . . Q Q
. . . . . .
. . . Q . Q . .
Q . . . . . .
. Q . . Q . . .
. . . . . .
. . . . . .
```

Final Board (Steepest-Ascent Hill Climbing):

```
. Q . . . . . .
. . . . . . . Q
. . . . . Q . .
. . . Q . . . .
Q . . . . . .
. . . . . Q .
. . . . Q . . .
. . Q . . . . .
```

Time Taken: 0.00599217414855957 seconds
Search Cost: 336

Final Board (Genetic Algorithm):

```
. . . . . Q . .
. . Q . . . . .
. . . . . Q .
. . . Q . . . .
Q . . . . . .
```

. . . . . Q
. Q . . . . .
. . . . Q . . .

Time Taken: 0.44840073585510254 seconds
Search Cost: 7900
The GA final board is a solution.

Conclusion

The project successfully implemented and compared two local search algorithms for solving the n-Queens problem. The Steepest-Ascent Hill Climbing algorithm, while faster and with lower search costs, was less effective in finding solutions due to its susceptibility to local optima. The Genetic Algorithm, though more computationally intensive, provided better results in terms of success rate, showcasing its strength in global search and exploration capabilities. Future work may include experimenting with different parameter settings for the Genetic Algorithm or exploring other advanced local search algorithms for more efficient and effective solutions.